

Claims

1. (Previously Presented) A method for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the method comprising:

transforming a first C/C++ program having a first multi-tasking property, wherein the first multi-tasking property comprises a property relating to a preemptive multitasking model, into a data structure; transforming the data structure to include an explicit multi-tasking transfer of control command;

optimizing the data structure to reduce an amount of program state that is saved at a transfer of control; and

generating a second C/C++ program having a second multi-tasking property, wherein the second multi-tasking property comprises a property relating to a run-to-completion model, using the optimized data structure

2. (Original) The method of claim 1, wherein the data structure further comprises a syntax tree.

3. (Original) The method of claim 2, wherein the step of transforming the data structure to include an explicit multi-tasking transfer of control command further comprises:
- converting the syntax tree to a continuation-passing style (CPS).
4. (Cancelled)
5. (Original) The method of claim 1, wherein the first program having a first multi-tasking property operates using a first program language and the second program having a second multi-tasking property also operates using the first program language.
6. (Previously Presented) A system for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the system comprising:
- a data structure transformer for transforming a first C/C++ program having a first multi-tasking property into a data structure, wherein the first multi-tasking

property comprises a property relating to a preemptive multitasking model;

a multi-tasking transformer for transforming the data structure to include an explicit multi-tasking transfer of control command;

a program state optimizer for optimizing the data structure to reduce an amount of program state that is saved at a transfer of control; and

a program generator for generating a second C/C++ program having a second multi-tasking property using the optimized data structure, wherein the second multi-tasking property comprises a property relating to a run-to-completion model.

7. (Original) The system of claim 6, wherein the data structure further comprises a syntax tree.

8. (Original) The system of claim 7, wherein the multi-tasking transformer further comprises:

a converter for converting the syntax tree to a continuation-passing style (CPS).

9. (Cancelled)

10. (Original) The system of claim 6, wherein the first program having a first multi-tasking property operates using a first program language and the second program having a second multi-tasking property also operates using the first program language.

11. (Previously Presented) An article of manufacture for transforming a C/C++ program having a first multi-tasking property to a C/C++ program having a second multi-tasking property, the article of manufacture comprising:

at least one processor readable carrier; and  
instructions carried on the at least one carrier;  
wherein the instructions are configured to be  
readable from the at least one carrier by at  
least one processor and thereby cause the at  
least one processor to operate so as to:

transform a first C/C++ program having  
a first multi-tasking property into a  
data structure, wherein the first

multi-tasking property comprises a  
property relating to a preemptive  
multitasking model;

transform the data structure to include an  
explicit multi-tasking transfer of  
control command;

optimize the data structure to reduce an  
amount of program state that is saved  
at a transfer of control; and

generate a second C/C++ program having a  
second multi-tasking property using the  
optimized data structure, wherein the  
second multi-tasking property comprises  
a property relating to a run-to-  
completion model.

12. (Previously Presented) A processor readable medium  
for providing instructions to at least one processor  
for directing the at least one processor to:  
transform a first C/C++ program having a first multi-  
tasking property into a data structure;

transform the data structure to include an  
explicit multi-tasking transfer of control  
command;  
optimize the data structure to reduce an amount  
of program state that is saved at a transfer  
of control; and  
generate a second C/C++ program having a second  
multi-tasking property using the optimized  
data structure.

13. (Previously Presented) A signal embodied in a  
carrier wave and representing sequences of  
instructions which, when executed by at least one  
processor, cause the at least one processor to  
transform a program having a first multi-tasking  
property to a program having a second multi-tasking  
property by performing the steps of:  
transforming a first C/C++ program having a first  
multi-tasking property into a data structure;  
transforming the data structure to include an  
explicit multi-tasking transfer of control  
command;

optimizing the data structure to reduce an amount  
of program state that is saved at a transfer  
of control; and  
generating a second C/C++ program having a second  
multi-tasking property using the optimized  
data structure.